

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 23.M07-мм

Логистический портал Cargotime.ru:
оптимизация инструментов для оценки
сроков прибытия контейнеров на основе
исторических данных и текущего
местоположения судна

СЧЕРЕВСКИЙ Виктор Максимович

Отчёт по производственной практике
в форме «Производственное задание»

Научный руководитель:
доцент кафедры информатики СПбГУ, к. т. н., Абрамов М.В.

Консультант:
м. н. с. лаб. ПИИ СПб ФИЦ РАН, Есин М.С.

Санкт-Петербург
2024

Оглавление

Введение	3
1. Постановка задачи	5
2. Правовые аспекты парсинга	6
3. Обзор сервиса	7
3.1. Архитектура сервиса трекинга	7
3.2. Используемые технологии	8
3.3. Принцип работы	10
4. Автоматизация сбора данных	12
4.1. Разработка сервиса для сбора данных	12
4.2. Результаты	13
5. Валидация статусов	15
5.1. Классификация статусов	15
5.2. Разработка инструмента валидации	16
6. Оптимизация парсинга	19
6.1. Идея и реализация подхода	19
6.2. Результаты	21
Заключение	23
Список литературы	25

Введение

Логистика – это ключевой элемент современной экономики. Она охватывает широкий спектр задач: от управления запасами до доставки товаров конечным потребителям. В условиях глобализации и растущей сложности цепочек поставок мониторинг перевозок, в частности, отслеживание перемещения контейнеров морских линий, приобретает критическую значимость [23, 12]. Эта возможность делает процесс доставки более прозрачным и предсказуемым для клиентов, а также позволяет оперативно реагировать на такие проблемы, как повреждение или кража контейнеров, что, в свою очередь, повышает надежность перевозки [26].

При этом одной из ключевых проблем остается низкая точность прогнозирования сроков доставки. По данным исследований [11], менее 60% контейнерных перевозок в мире завершаются в указанные компанией сроки, что приводит к сбоям в планировании, увеличению затрат и снижению эффективности логистических процессов. Данные за июнь 2024 года показывают, что среднее время задержки контейнерных перевозок составляет более 5 суток [11]. Проблема обостряется при планировании мультимодальных маршрутов, где необходимо учитывать как морские, так и наземные виды транспорта.

Существующие системы мониторинга и оценки сроков доставки, в том числе и крупные западные аналоги [10, 17, 5], характеризуются значительными ограничениями. Они либо не используют данные о текущем местоположении судна, либо игнорируют исторические сведения о задержках и иных особенностях маршрутов. Это приводит к невозможности предсказать отклонения, вызванные, например, сезонной перегруженностью портов или изменениями в инфраструктуре. Вследствие этого системы предоставляют лишь частичные решения, не позволяя эффективно управлять мультимодальными цепочками поставок.

На фоне данных ограничений портал Cargotime.ru [27] выступает как активно развивающийся цифровой инструмент, цель которого – предоставить клиентам грузовых компаний простой и доступный способ

получения информации о перевозке грузов. Среди его ключевых сервисов – инструменты для оценки сроков прибытия контейнеров, включая интерактивную карту с функцией отслеживания маршрута контейнера. Эти инструменты востребованы благодаря возможности самостоятельно оценивать сроки доставки независимо от предоставленных перевозчиками данных, что особенно важно для минимизации рисков в логистике.

Однако, чтобы обеспечить высокую точность и надежность таких инструментов, необходимо преодолеть ряд проблем, связанных с качеством и полнотой данных. Например, сбор информации о координатах портов и местоположении судов часто затруднен из-за недостаточной автоматизации процесса, а данные о статусах перевозки требуют дополнительной обработки, поскольку различные компании используют свои уникальные обозначения.

Настоящая работа направлена на решение данных проблем путем оптимизации существующих инструментов. Таким образом, она представляет собой шаг к созданию комплексной системы прогнозирования, которая обеспечит высокую точность и надежность мультимодальных логистических операций.

1. Постановка задачи

Целью данной работы является повышение эффективности и стабильности работы инструментов для оценки сроков прибытия контейнеров за счет автоматизации сбора исторических данных и внедрения системы валидации состояний перевозки. Для её выполнения были поставлены следующие задачи:

- Разработать сервис, автоматизирующий сбор часто запрашиваемых портов и судов на основе данных из логов;
- Реализовать редактируемый валидирующий механизм, моделирующий систему переходов статусов перевозки контейнеров;
- Реализовать эффективный алгоритм для массового сбора данных о портах и судах из открытых источников;
- Внедрить полученное решение в существующий продукт.

2. Правовые аспекты парсинга

Парсинг, или “веб-скрейпинг”, представляет собой процесс автоматизированного сбора данных с веб-ресурсов. В российском законодательстве это понятие четко не определено, что создает правовую неопределенность. Однако, если сбор данных ограничивается открытыми и свободно распространяемыми источниками, такие действия могут быть признаны соответствующими ст. 29, п. 4 Конституции РФ [2], которая гарантирует право на поиск, получение и распространение информации. При этом сбор данных должен осуществляться с учетом законодательства о защите персональных данных [28], государственной тайне и интеллектуальной собственности [1].

Легитимность парсинга также во многом зависит от соблюдения локальных правил ресурса, с которого собираются данные. Многие веб-сайты устанавливают в пользовательских соглашениях запрет на автоматизированный сбор данных. Если пользователь регистрируется на сайте и принимает условия использования, он заключает договор, который подписывает простой электронной подписью. В случае нарушения условий договора, например, при попытке обхода защитных механизмов с использованием автоматизированных инструментов, эти действия могут повлечь ответственность в отношении разработчика или пользователя парсера [13].

Парсеры, собирающие данные о контейнерах из открытых источников, не противоречат существующему законодательству РФ, а также не подразумевают авторизацию в системе компании и принятие пользовательского соглашения. Следовательно, привлечь к ответственности пользователя или разработчика такого парсера на основании вышеуказанных положений невозможно.

3. Обзор сервиса

В данном разделе будут приведены высокоуровневое описание той части сервиса трекинга контейнеров, к которой непосредственно относятся инструменты для оценки сроков доставки, а также краткий обзор используемых технологий и принцип работы сервиса.

3.1. Архитектура сервиса трекинга

Все компоненты портала Cargotime.ru, включая сервис трекинга контейнеров, реализованы с использованием микросервисной архитектуры. Такой подход обеспечивает модульность, гибкость и масштабируемость системы, что позволяет балансировать нагрузку на отдельные микросервисы, а также обеспечивает отказоустойчивость всего сервиса в случае выхода из строя какой-то из его частей.

Часть архитектуры сервиса трекинга контейнеров, связанная с инструментами для оценки сроков доставки, выглядит следующим образом:

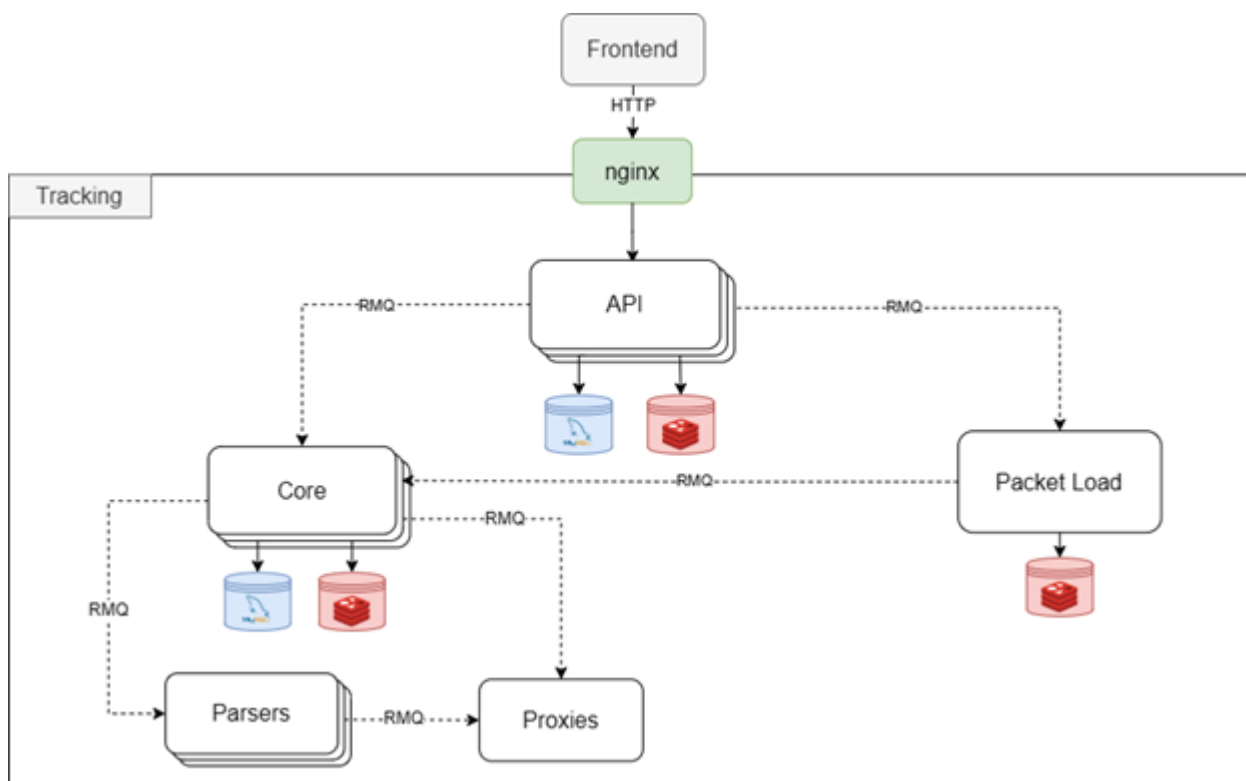


Рис. 1: Часть архитектуры сервиса трекинга

На схеме сплошными стрелками с меткой “HTTP” обозначено взаимодействие по протоколу HTTP, пунктирными стрелками – взаимодействие с помощью брокера сообщений RabbitMQ по протоколу AMQP. Эта технология играет ключевую роль в обеспечении отказоустойчивости системы, поскольку предоставляет гарантии доставки сообщений, а также помогает распределять нагрузку между репликами микросервисов. Синим и красным цветами обозначены база данных MySQL и хранилище Redis соответственно. Более подробное описание этих технологий будет приведено в следующем разделе.

Описание компонентов:

- **Core** – ядро парсеров, в котором обрабатываются все запросы к источникам. Данный сервис реплицируется.
- **API** – промежуточный сервис между веб-интерфейсом и парсерами. В нём агрегируется и кэшируется метаданная о компаниях и направляется клиенту вместе с ответами **Core**.
- **Packet Load** – сервис, необходимый для обработки больших групп трек-номеров, которые клиенты могут загрузить через специальный интерфейс на сайте.
- **Proxies** – оптимально распределяет динамические и статические порты для парсинга.
- **Parsers** – сервис, содержащий дополнительные инструменты для парсинга открытых источников.
- **Frontend** – веб-сайт, предоставляющий клиентский интерфейс для взаимодействия с инструментами портала, такими, как интерактивная карта с маршрутом контейнера, и другими.

3.2. Используемые технологии

Ниже перечислены основные программные технологии, используемые в сервисе трекинга контейнеров:

- **Node.js** [16] – программная платформа с открытым исходным кодом, предназначенная для разработки серверных приложений на языке JavaScript.
- **TypeScript** [24] – строго типизированный язык программирования с открытым исходным кодом, компилирующийся в JavaScript и расширяющий его возможности. Повсеместно применяется в проекте для построения серверных приложений.
- **NestJS** [15] – современный фреймворк с открытым исходным кодом для создания производительных и масштабируемых серверных приложений на Node.js. Поддерживает разработку на языках JavaScript и TypeScript.
- **RabbitMQ** [20] – открытая система обмена сообщениями (брокер сообщений), реализующая протокол AMQP. С помощью RabbitMQ производится обмен данными между ядром парсеров, API трекинга и другими сервисами.
- **Redis** [21] – высокопроизводительная in-memory (в оперативной памяти) база данных с доступным исходным кодом (source-available). Используется для кэширования данных в сервисе ядра парсеров, API трекинга и сервиса пакетной загрузки.
- **ELK (Elasticsearch, Logstash, Kibana)** [9] – стек технологий для поиска, анализа и визуализации логов. Используется для поиска информации о новых локациях среди ответов парсеров.
- **MySQL** [14] – свободная реляционная СУБД. Используется в сервисе ядра парсеров для хранения информации о компаниях, портах и др.
- **Docker** [8] – открытая платформа для разработки, доставки и запуска приложений в контейнерах. Также используется Docker Compose для автоматизированного развертывания контейнеров в единой сети.

- **Playwright** [18] – библиотека для автоматизированной работы с веб-сайтами. Поддерживает эмуляцию различных браузеров.
- **Puppeteer** [19] – еще один инструмент для эмуляции работы браузера. Ориентирован преимущественно на Chromium [4].

3.3. Принцип работы

На изображениях ниже показан возможный сценарий работы пользователя с интерактивной картой для отслеживания контейнера:

Рис. 2: Ввод данных для трекинга

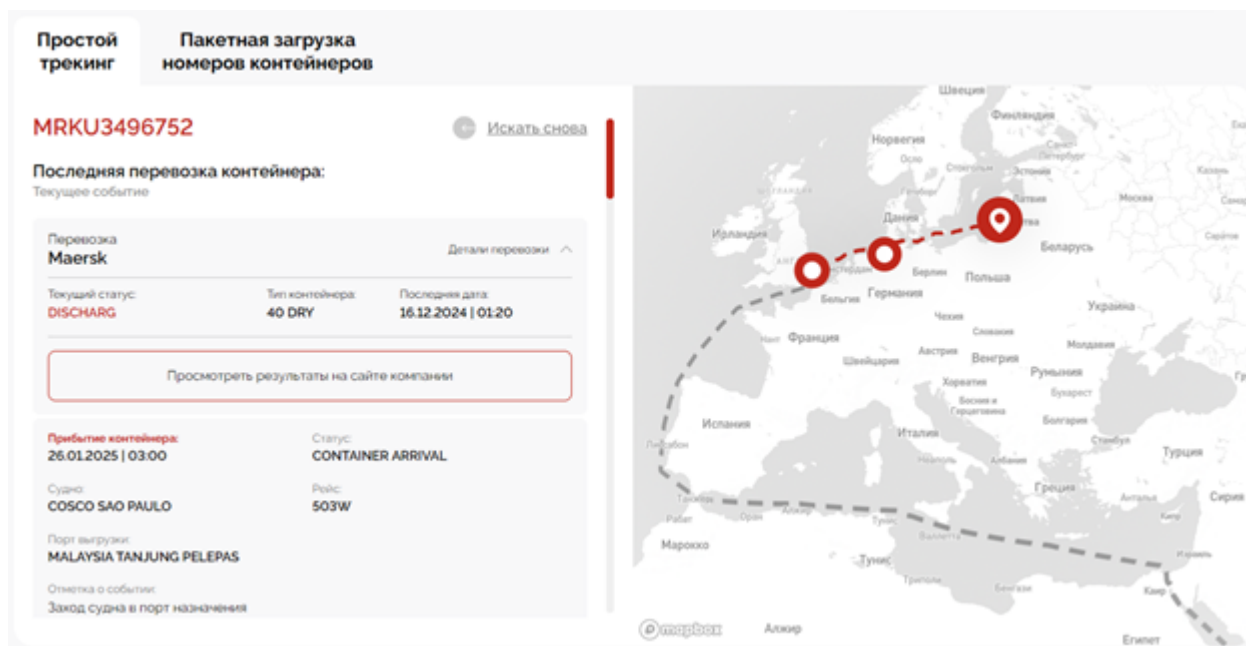


Рис. 3: Результат поиска информации о контейнере

После ввода данных контейнера, построение маршрута следования осуществляется в 3 основных этапа:

1. **Получение данных о локациях контейнера.** Отправляется запрос к парсеру для получения локаций контейнера, т.е. локаций, по которым он будет следовать.
2. **Обработка ответа парсера.** Локации в ответе могут быть расположены в неправильном порядке, поэтому их нужно упорядочить в соответствии со статусами перевозки.
3. **Построение и визуализация маршрута.** На завершающем этапе маршрут отрисовывается на основе упорядоченных координат локаций. Это обеспечивает корректное отображение пути следования контейнера на карте.

Этот алгоритм подчеркивает ключевую роль данных о локациях и статусах перевозки. Без этой информации невозможно корректно построить маршрут, что делает ее сбор и обработку критически важными.

4. Автоматизация сбора данных

4.1. Разработка сервиса для сбора данных

Для обеспечения точности прогнозирования маршрутов и сроков доставки контейнеров необходимо наличие актуальной и полной информации о наиболее часто запрашиваемых локациях и судах. В рамках решения этой задачи был разработан сервис **Elastic Utils**, предназначенный для автоматизации сбора и обработки данных из логов с использованием **Elasticsearch**.

Сервис **Elastic Utils** выполняет следующие ключевые функции:

- **Поиск данных в логах.** Сервис интегрирован с Elasticsearch, что позволяет осуществлять поиск по логам запросов пользователей. Таким образом, собираются данные о локациях и судах, которые чаще всего фигурируют в запросах.
- **Агрегация и обработка данных.** После извлечения данных из логов сервис агрегирует информацию, устраняя дублирование и сортируя локации по частоте их использования.
- **Предоставление данных в удобном формате.** Результаты работы сервиса предоставляются в формате CSV-файлов для последующей обработки специалистами. Также эти данные могут запрашиваться сервисом ядра в автоматическом режиме с заданной периодичностью для обновления информации.

Часть архитектуры сервиса трекинга после внедрения сервиса **Elastic Utils** выглядит, как представлено на рисунке ниже:

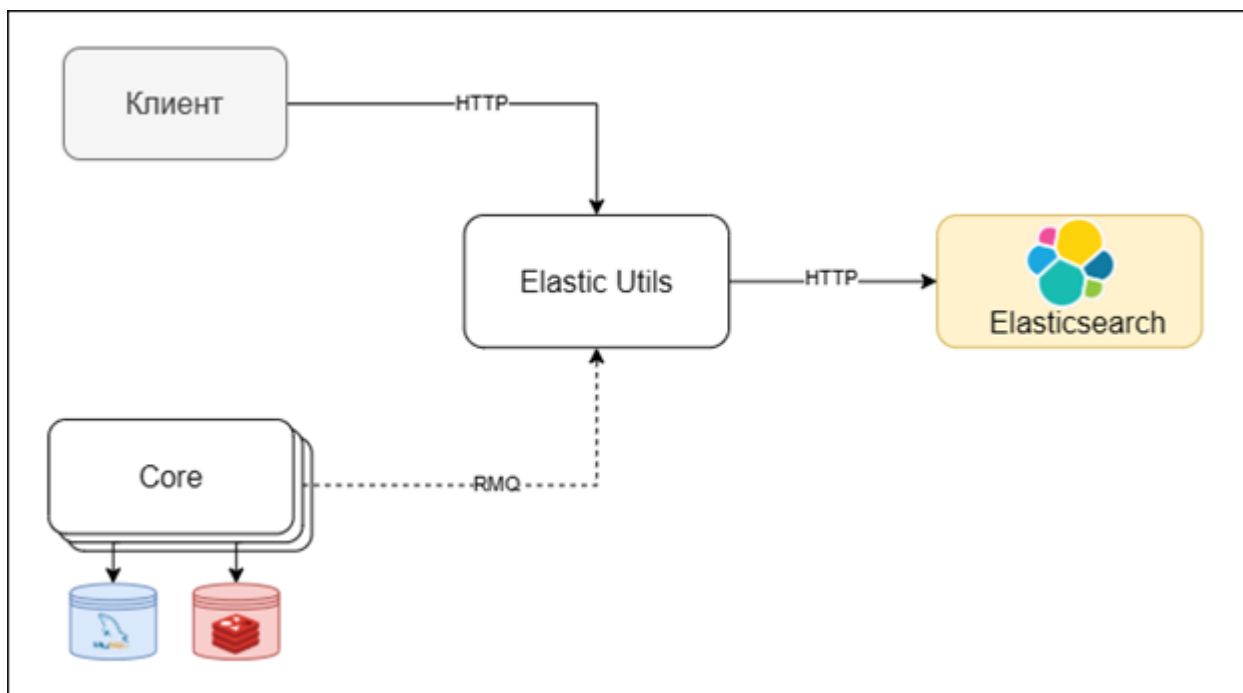


Рис. 4: Внедрение Elastic Utils

4.2. Результаты

В результате работы сервиса **Elastic Utils** удалось добиться следующих улучшений:

- Были определены 25 наиболее популярных портов, которые встречаются в 70% запросов.
- Улучшены показатели по двум метрикам (см. таблицу 1):
 - **Покрытие summary локаций** – рассчитывается, как доля известных summary локаций относительно числа всех summary локаций, когда-либо встреченных в ответах парсеров.
Summary локации: пункт отправления, порт загрузки, порт выгрузки, пункт назначения. Этот показатель характеризует полноту отображения наиболее значимой информации.
 - **Покрытие port локаций** – аналогичная метрика для всех морских портов в маршруте контейнера. Отражает точность построения маршрутов.

Метрика	Исходный показатель	Полученный показатель	Прирост
Покрытие summary локаций	79,04%	93,64%	+14,60%
Покрытие port локаций	82,60%	94,70%	+12,10%

Таблица 1: Прирост показателей покрытия локаций

Внедрение сервиса **Elastic Utils** также позволило существенно сократить трудозатраты на обновление информации о портах и судах, устранив необходимость ручного поиска и обработки данных.

5. Валидация статусов

5.1. Классификация статусов

Для корректного построения маршрута необходимо понимать значение статусов перевозки и возможные переходы между ними, чтобы верно упорядочить локации в пути следования контейнера. Однако не существует какого-то единого формата именования статусов перевозки, поэтому каждая компания имеет собственные уникальные названия. В связи с этим командой Cargotime была разработана собственная классификация статусов с целью унифицировать всевозможные наименования статусов различных компаний. За основу был взят мировой стандарт DCSA [7], которому уже следуют многие крупные перевозчики [6]. Всего было выделено 18 статусов:

- CEP – Отправка порожнего контейнера;
- CPS – Забор грузеного контейнера;
- CGI – Прием грузеного контейнера;
- CLL – Погрузка в порту отправления;
- VDL – Выход судна из порта отправления;
- VAT – Заход судна в транзитный порт;
- CDT – Выгрузка в транзитном порту;
- TSD – Задержка в пути;
- CLT – Погрузка в транзитном порту;
- VDT – Выход судна из транзитного порта;
- VAD – Заход судна в порт назначения;
- CDD – Выгрузка в порту назначения;

- CGO – Выдача грузевого контейнера;
- CDC – Доставка грузевого контейнера;
- CER – Возврат порожнего контейнера;
- LTS – Перевозка наземным транспортом;
- BTS – Перевозка водным транспортом;
- UNK – Не определен.

Эта классификация позволяет выстраивать маршрут следования контейнера независимо от компании-перевозчика, предоставляющей информацию. Однако проверка корректности этой классификации была затруднительной ввиду отсутствия каких-либо валидирующих инструментов.

5.2. Разработка инструмента валидации

С целью повысить информативность отчетов о корректности построенных маршрутов, а также иметь возможность более сознательно подходить к редактированию классификации статусов и переходов между ними, было принято решение разработать систему, представляющую унифицированные статусы в виде машины состояний, а также позволяющую эту машину состояний редактировать удобным способом.

Разработанный инструмент основан на функциональности библиотеки **XState** [25], которая позволяет работать с машиной состояний в TypeScript. Выбор этой библиотеки во многом обусловлен тем, что она предлагает удобную интеграцию с платформой для визуального моделирования **Statel.ai** [22]. Формат описания машины состояний полностью совместим с этим редактором, что означает, что пользователь может описать новую машину состояний или отредактировать существующую в этой платформе, сохранить ее в json формате, а затем загрузить в нужный сервис. Ниже проиллюстрирован пример построения машины состояний в редакторе Statel.ai:

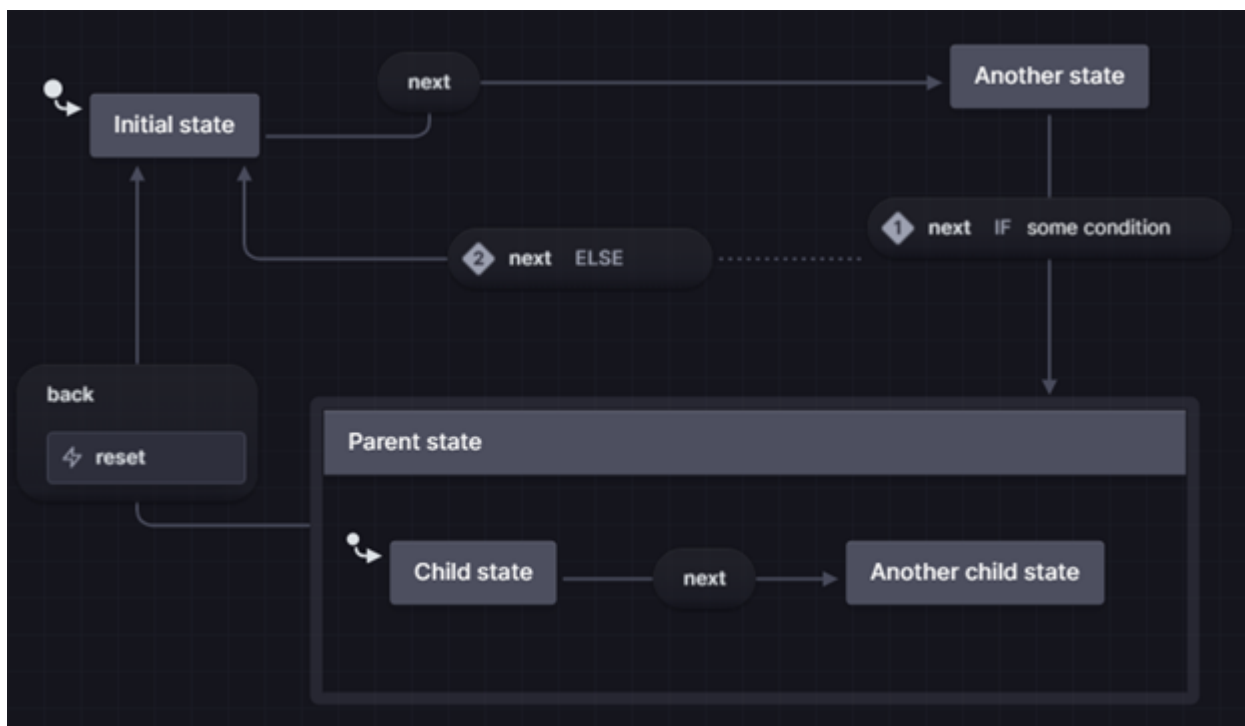


Рис. 5: Редактор Stately.ai

Таким образом, разработанный инструмент выполняет 3 основные функции:

- **Валидация статусов.** Системе можно передавать статусы и переходы между ними, а в ответе будет информация о том, являются ли эти переходы валидными, в соответствии с заданной машиной состояний.
- **Удобное редактирование машины состояний.** В комбинации с платформой Stately.ai, специалисты могут редактировать машину состояний без необходимости написания кода.
- **Построение диаграммы переходов.** Инструмент также позволяет обработать данные о перевозках контейнеров и построить на их основе диаграмму переходов статусов. Это позволяет визуально оценить корректность классификации статусов и предпринять действия в случае обнаружения ошибок.

На данном этапе инструмент работает в тестовом режиме в сервисе **Elastic Utils**. К текущему моменту не было собрано достаточное

количество статистических данных для подробного анализа работы инструмента. Тем не менее, был проведен небольшой эксперимент: была построена общая диаграмма переходов для последних 1000 перевозок. На этой диаграмме были переходы практически между всеми статусами, что явно свидетельствует о том, что многие статусы были классифицированы неверно и требуют пересмотра.

6. Оптимизация парсинга

6.1. Идея и реализация подхода

Сервис **Parsers** является важным компонентом портала Cargotime, предоставляя данные о портах и судах из открытых источников. Этот сервис построен с использованием библиотеки Playwright, которая позволяет эмулировать действия браузера для работы с веб-страницами. В рамках оптимизации сервиса были предприняты шаги, направленные на повышение его производительности и эффективности использования ресурсов.

Сервис реплицируется для обработки большого количества страниц, что позволяет распределять нагрузку. Однако, для оптимального использования ресурсов на одной физической машине, важно учитывать природу операций ввода-вывода (I/O-bound), которые выполняются при взаимодействии с браузером через библиотеку Playwright. Playwright создаёт страницы в браузере Chromium, которые способны работать в отдельных потоках, в связи с чем возникает идея воспользоваться свойством асинхронности Node.js для передачи нагрузки эмулятору браузера. На рисунке ниже проиллюстрирована схема такого взаимодействия:

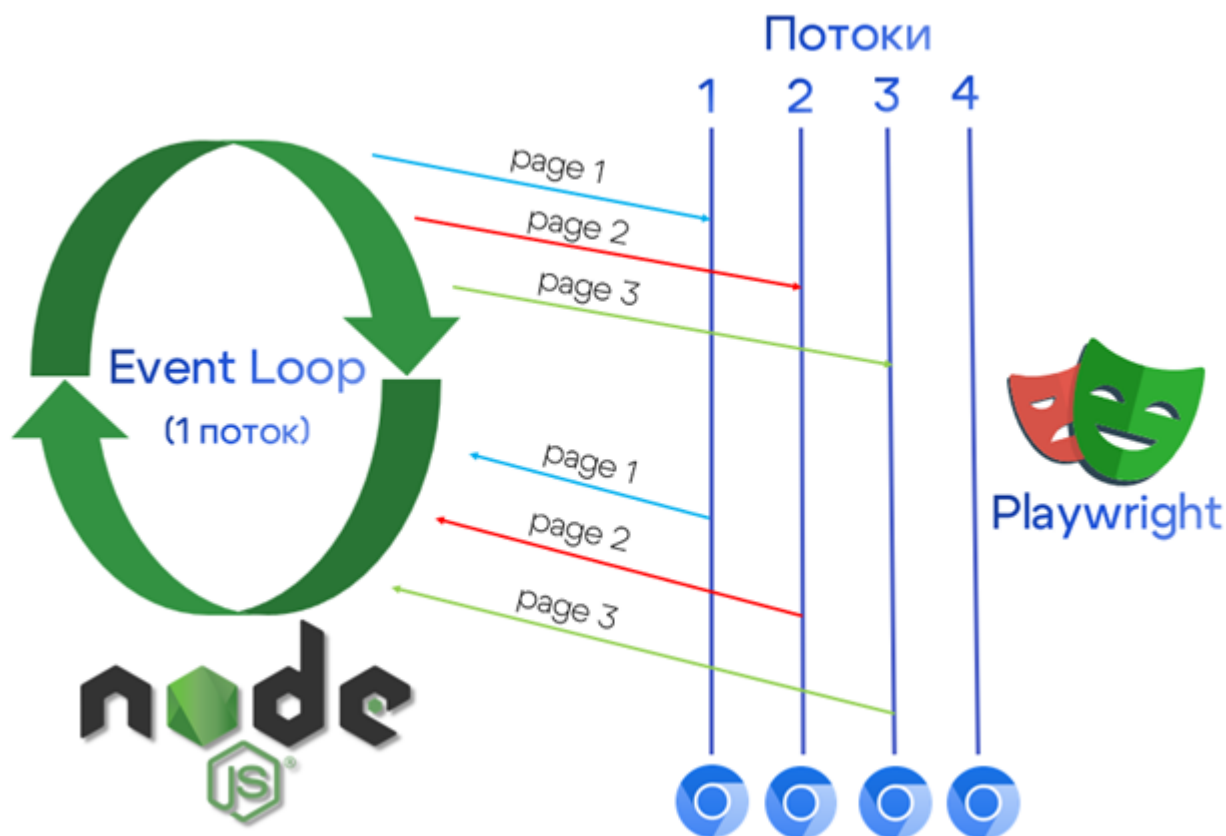


Рис. 6: Асинхронное взаимодействие с Playwright

Такой подход, в теории, должен увеличивать производительность. Однако при этом нужно учесть риски, связанные с чрезмерной частотой запросов, которые могут привести к появлению CAPTCHA [3] на целевых сайтах. Для контроля асинхронности был реализован механизм ограничения количества одновременных запросов и обработки ошибок, включающий повторные попытки в случае сбоев. Реализация подхода представляет собой метод `mapConcurrently()`, асинхронно обрабатывающий массив элементов и принимающий в качестве параметров:

- `items` – массив элементов, подлежащих асинхронной обработке;
- `mapper` – вызываемая функция-обработчик;
- `concurrent` – “степень асинхронности” (максимальное число асинхронных вызовов);
- `timeout` – лимит по времени на работу функции;

- `retryCount` – количество повторных попыток вызова функции.

6.2. Результаты

Применение подхода на практике продемонстрировало заметное улучшение производительности сервиса парсинга. В качестве эксперимента в парсер, запущенный на одной физической машине, отправлялись запросы на обработку информации со страниц для множества портов. Измерение каждого случая проводилось 20 раз, затем результат усреднялся. Влияние качества интернет-соединения в эксперименте не учитывалось, поскольку время отклика страниц и объем передаваемого трафика незначительны относительно затрат на обработку страниц.

Программно-аппаратная конфигурация приведена в таблице ниже:

Процессор	Intel Core i7-8750H CPU 2.20GHz (Coffee Lake), 1 CPU, 12 logical and 6 physical cores
Объем ОЗУ	16 GB
Операционная система	Windows 11 23H2
Версия Node.js	22.9.0
Версия Playwright	1.43.1

Таблица 2: Конфигурация

Результаты измерений:

Кол-во портов в запросе	Среднее время \pm ст. откл. (1 поток, с)	Среднее время \pm ст. откл. (6 потоков, с)
6	53 ± 8	17 ± 8
12	112 ± 11	39 ± 18
24	208 ± 18	81 ± 29
60	424 ± 22	158 ± 46
120	864 ± 38	311 ± 65

Таблица 3: Результаты измерений времени работы парсера

Оптимальными значениями параметров для заданной конфигурации оказались следующие величины:

- “Степень асинхронности” – 6;
- Кол-во повторных попыток – 5;
- Таймаут – 30 с.

Именно результаты измерений с этими значениями параметров приведены в правой части таблицы 3.

Полученные результаты эксперимента свидетельствуют о значительном приросте производительности при использовании асинхронного подхода, что подтверждает эффективность выбранного метода. Также стоит отметить универсальность данного подхода в том смысле, что его можно применять не только для получения информации от парсеров портов и судов, но также и для массовых запросов к любым другим парсерам, реализованным с помощью библиотеки Playwright.

Заключение

Целью работы являлось повышение эффективности и стабильности работы инструментов для оценки сроков прибытия контейнеров за счет автоматизации сбора исторических данных и внедрения системы валидации состояний перевозки.

В ходе работы были выполнены все поставленные для достижения цели задачи, а именно:

- Автоматизирован сбор часто запрашиваемых портов и судов на основе данных из логов. Покрытие summary и port локаций увеличено на 14,6% и 12,1% соответственно.
- Реализован редактируемый валидирующий механизм, моделирующий систему переходов статусов перевозки контейнеров.
- Реализован эффективный алгоритм для массового сбора данных о портах и судах из открытых источников.
- Разработанные инструменты внедрены в портал **Cargotime**.

Эффективность инструментов для оценки сроков прибытия контейнеров была повышена за счет сведения к минимуму затрат ручного труда при сборе информации о портах и судах, увеличения покрытия популярных локаций, а следовательно, и доступных для построения маршрутов, а также благодаря увеличению скорости отклика парсеров портов и судов при массовых запросах. Стабильность этих инструментов была повышена за счет внедрения системы валидации статусов перевозки, позволяющей обнаруживать неточности во внутренней классификации статусов, корректировать их и тем самым минимизировать количество ошибок при построении маршрутов.

В дальнейшем также планируется выполнить следующие задачи:

- Автоматизировать регулярное обновление данных о популярных судах с целью снижения времени отклика.

- Оценить эффективность работы валидатора статусов. Провести анализ обнаруженных ошибок.

Список литературы

- [1] 272 статья Уголовного кодекса Российской Федерации. — URL: <https://ukodeksrf.ru/ch-2/rzd-9/gl-28/st-272-uk-rf> (дата обращения: 25 декабря 2024 г.).
- [2] 29 статья Конституции Российской Федерации. — URL: <https://main-law.ru/st29> (дата обращения: 25 декабря 2024 г.).
- [3] CAPTCHA. — URL: <http://captcha.net/> (дата обращения: 25 декабря 2024 г.).
- [4] Chromium. — URL: <https://www.chromium.org/Home/> (дата обращения: 25 декабря 2024 г.).
- [5] Container Tracking. — URL: <https://www.visiwise.co/tracking/container/> (дата обращения: 1 октября 2024 г.).
- [6] DCSA Standards. — URL: <https://dcsa.org/standards> (дата обращения: 25 декабря 2024 г.).
- [7] DCSA Track Trace Standard documentation. — URL: <https://dcsa.org/standards/track-and-trace/standard-documentation-track-and-trace> (дата обращения: 25 декабря 2024 г.).
- [8] Docker. — URL: <https://www.docker.com/> (дата обращения: 25 декабря 2024 г.).
- [9] ELK Stack. — URL: <https://www.elastic.co/elastic-stack> (дата обращения: 25 декабря 2024 г.).
- [10] Find the best Freight Quote. — URL: <https://www.searates.com/> (дата обращения: 1 октября 2024 г.).
- [11] Napag most reliable top-13 carrier in June 2024: Sea-Intelligence. — URL: <https://www.itln.in/shipping/>

[hapag-most-reliable-top-13-carrier-in-june-2024-sea-intelligence-infinitescroll=1](#) (дата обращения: 1 октября 2024 г.).

- [12] Impacts of a Tracking and Tracing System for Containers in a Port-Based Supply Chain / Jesús Muñuzuri, Luis Onieva, Alejandro Escudero, Pablo Cortés // [Brazilian Journal of Operations and Production Management](#). — 2016. — . — Vol. 13, no. 3. — P. 352–359. — URL: <http://dx.doi.org/10.14488/BJOPM.2016.v13.n3.a12>.
- [13] Krotov Vlad, Johnson Leigh, Silva Leiser. Legality and Ethics of Web Scraping // [Communications of the Association for Information Systems](#). — 2020. — Vol. 47. — P. 539–563. — URL: <http://dx.doi.org/10.17705/1CAIS.04724>.
- [14] MySQL. — URL: <https://www.mysql.com/> (дата обращения: 25 декабря 2024 г.).
- [15] NestJS. — URL: <https://nestjs.com/> (дата обращения: 25 декабря 2024 г.).
- [16] Node.js. — URL: <https://nodejs.org/en> (дата обращения: 25 декабря 2024 г.).
- [17] Online Container Tracking - Your GoComet Portal. — URL: <https://www.gocomet.com/online-container-tracking> (дата обращения: 1 октября 2024 г.).
- [18] Playwright. — URL: <https://playwright.dev/> (дата обращения: 25 декабря 2024 г.).
- [19] Puppeteer. — URL: <https://pptr.dev/> (дата обращения: 25 декабря 2024 г.).
- [20] RabbitMQ. — URL: <https://www.rabbitmq.com/> (дата обращения: 25 декабря 2024 г.).
- [21] Redis. — URL: <https://redis.io/> (дата обращения: 25 декабря 2024 г.).

- [22] Stately.ai. — URL: <https://stately.ai/> (дата обращения: 25 декабря 2024 г.).
- [23] [Tracking and Tracing of Global Supply Chain Network: Case Study from a Finnish Company](#) / Ahm Shamsuzzoha, Michael Ehlers, Richard Addo-Tengkorang, Petri Helo // Proceedings of the 23rd International Conference on Enterprise Information Systems. — SCITEPRESS - Science and Technology Publications, 2021. — P. 118–125. — URL: <http://dx.doi.org/10.5220/0010515401180125>.
- [24] TypeScript. — URL: <https://www.typescriptlang.org/> (дата обращения: 25 декабря 2024 г.).
- [25] XState. — URL: <https://xstate.js.org/> (дата обращения: 25 декабря 2024 г.).
- [26] Yue Zhongyun, Mangan John. A framework for understanding reliability in container shipping networks // [Maritime Economics and Logistics](#). — 2023. — . — Vol. 26, no. 3. — P. 523–544. — URL: <http://dx.doi.org/10.1057/s41278-023-00269-7>.
- [27] Логистический портал Cargotime. — URL: <https://cargotime.ru/> (дата обращения: 25 декабря 2024 г.).
- [28] Федеральный закон «О персональных данных». — URL: <http://letters.kremlin.ru/info-service/acts/9> (дата обращения: 25 декабря 2024 г.).